Web Database Integration

Wei Liu
School of Information
Renmin University of China
Beijing, 100872, China
gue2@ruc.edu.cn

Xiaofeng Meng
School of Information
Renmin University of China
Beijing, 100872, China
xfmeng@ruc.edu.cn

ABSTRACT

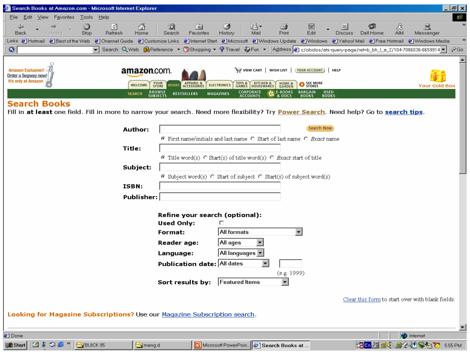
More and more accessible databases are available in the Web. In order to provide people a unified access to these Web databases and achieve information from them automatically, a comprehensive solution for Web database integration is proposed in this paper. After summarizing the research status in this area, the works which are the focus of my PhD thesis are presented.

1. INTRODUCTION

With the rapid development of Web, more and more accessible databases are available in the Web. Such databases are usually called Web database (or WDB in short) by researchers. From this angle, the Web can be divided into two parts: Surface Web and Deep Web. The Surface Web refers to the static Web pages which can be crawled and indexed by popular search engines, while the Deep Web refers to the contents stored in Web databases and published by dynamic Web pages.

The abundant information stored in Web databases is "hided" behind the query interfaces in Web pages. This means that the main approach people access Web databases is through their query interfaces. Figure 1 gives the query interface provided by Amazon which is a very popular e-commerce Web site.

According to the survey[1] released by UIUC in 2004, there are more than 300,000 Web databases and 450,000 query interfaces available at that time, and the two figures are still increasing quickly. Besides the scale of Web databases, the contents in Web databases are spanning well across all topics. Some Deep Web portal services provide Deep Web directories which classify Web databases in some taxonomies. For example, CompletePlanet[2], the biggest Deep Web directory, has collected more than 7,000 Web databases and classified them into 42 topics. Combing the above two aspects, we can conclude that theses Web databases are just like a huge repository and provide people a great opportunity to get their desired information.


Search Books at Amazon.com - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss Dell Home AIM Messenger
Links Hotmail Best of the Web Channel Guide Customize Links Internet Start Microsoft Windows Update Windows Yahoo! Mail Free Hotmail Windows Media
Search: Web Reference Shopping Travel Fun Address c/obidos/ats-query-page/ref=b_bh_l_a_2/104-7086036-8659914 Go
Amazon Exclusive!! Order a Segway now! It's only at Amazon
amazon.com.
VIEW CART WISH LIST YOUR ACCOUNT HELP
WELCOME YOUR STORE BOOKS APPAREL & ACCESSORIES ELECTRONICS TOYS & GAMES KITCHEN & HOUSEWARES HOME & GARDEN SEE MORE STORES
Your Gold Box
SEARCH BROWSE SUBJECTS BESTSELLERS MAGAZINES CORPORATE ACCOUNTS E-BOOKS & DOCS BARGAIN BOOKS USED BOOKS
Search Now
Done Internet
Start BUICK 95 meng.d Microsoft PowerPoin... Search Books at ... 5:55 PM

Search Books

Fill in **at least** one field. Fill in more to narrow your search. Need more flexibility? Try Power Search. Need help? Go to search tips.

Author:
● First name/initials and last name ○ Start of last name ○ *Exact* name
Title:
● Title word(s) ○ Start(s) of title word(s) ○ *Exact* start of title
Subject:
● Subject word(s) ○ Start of subject ○ Start(s) of subject word(s)
ISBN:
Publisher:

Refine your search (optional):
Used Only: □
Format: All formats
Reader age: All ages
Language: All languages
Publication date: All dates (e.g. 1999)
Sort results by: Featured Items

Clear this form to start over with blank fields.

Looking for Magazine Subscriptions? Use our Magazine Subscription search.

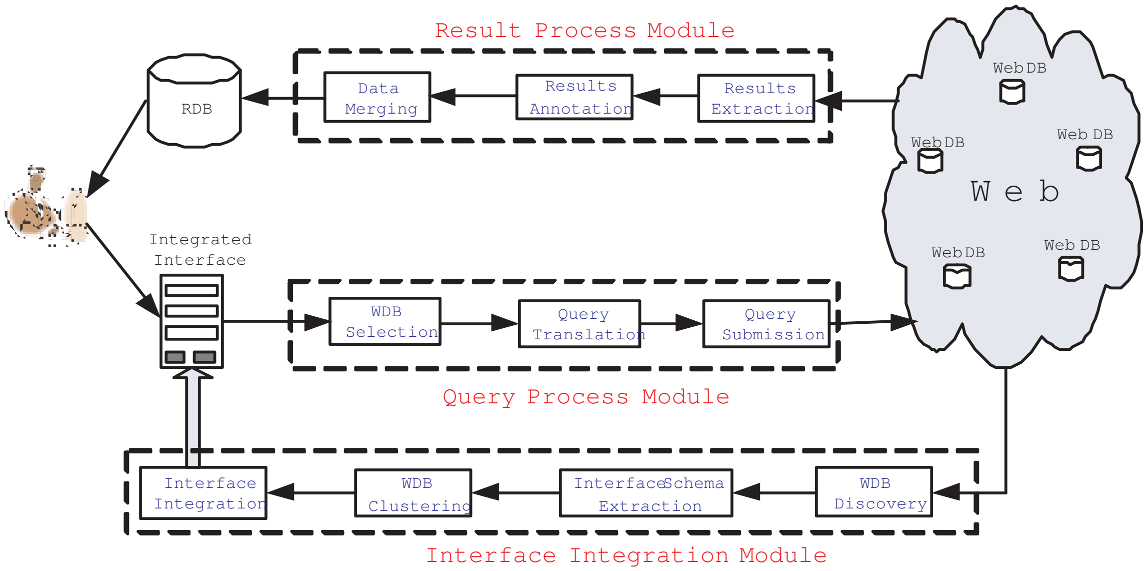
Figure 1: The query interface of Amazon

With proliferation of Web databases, it is not only an opportunity but also a challenge for people. At present, people access to Web databases mainly by manual approach, and his will bring an overhead problem.

Here is an example to explain the problem. Suppose Jane wants buy a book on Java. There are several tasks she has to complete. First, she must find the Web sites which sell books. If she wants save money, more Web sites are needed to compare. Second, she fills the query interfaces with an appropriate query (for example, fill book title with "think in java") and submits them. Third, when the Web pages contain query results returned (these Web pages are called response pages generally), she browses them in turn and chooses the best book. The whole process is time-consuming. Maybe Jane will spend half a day for this. Therefore, the challenge of manual approach is people often have difficulties in first finding the right sources and then querying over them.

It is impending and compulsory to integrate Web databases and to provide people a unified access to them and achieve information automatically. Web databases integration can be considered as the heterogeneous data source integration in Web context. The traditional heterogeneous data source integration generally focuses on the heterogeneity and autonomy of data sources. According to my investigation, Web databases also have four distinct characteristics which are different to other heterogeneous data sources:

Proceedings of the VLDB2006 Ph.D. Workshop
Seoul, Rep of Korea, 2006



ss

Figure 2: A comprehensive solution for Web database integration

- ***Scale:*** There are myriads of Web databases in Web, and even under a special topic the quantity of Web databases is still striking.
- ***Dynamic:*** First, Web databases are very sparsely distributed in Web, and they appear and disappear endlessly. So searching for appropriate Web databases in Web is really like looking for a few needles in a haystack. Second, the contents in Web databases are usually updated frequently. Especially in some topics, such as airline and job, everyday a batch of new contents will be added to Web databases and the outdated part will be removed. So the information in Web databases is "ever" not "forever" to you.
- ***Access through query interfaces:*** Due to the peculiar access approach, the schema of a Web database can not be captured directly. We can only infer the schema from their query interfaces and response pages.
- ***Heterogeneity:*** The query interfaces and response pages are designed by different persons and there are no design standards to follow. Even in the same topic, the query interfaces and response pages are often very dissimilar.

In a word, the research on Web database integration aims to help people make use of the abundant information in Web databases effectively and efficiently. But due to the distinct characteristics of Web databases, there are many challenging research issues in this area.

My PhD thesis is focusing on building a Web database integration system and addressing several challenging issues in this area. In this paper a comprehensive solution for Web database integration is presented and my current and future research works in this area is indicated .

There is a fact which should not be neglected. Some Web sites provide Web Services for their Web databases, and people can use a customized program to access Web databases. But this approach has two limitations: first, only a small portion of Web sites provide Web Services for their Web databases; second, this approach must depend on a customized program, and this is not an easy thing for common users. So in this paper we focus on the popular approach of accessing Web databases through the query interfaces in Web pages.

The rest of this paper is organized as follows. Section 2 gives the solution for Web database integration; Section 3 summarizes the research status in this area; Section 4 presents the works we are focusing now and will focus in the future; Section 5 is the conclusion.

2. A SOLUTION FOR WDB INTEGRATION

In this section, a comprehensive solution for Web database integration is proposed, which is the pursuit in my PhD track. Figure 2 is the architecture of the solution. This solution includes three primary modules: integrated interface generation module, query processing module and results processing module.

Integrated interface generation module: Produce an integrated interface over the query interfaces of the Web databases to be integrated. There are four components in this module. The functions of them are described as following:

- *Web database discovery:* Search Web sites which have Web databases behind, and identify the query interfaces among the Web pages in these Web sites.
- *Query interface schema extraction:* Extract the attributes in query interfaces (such as "Title" and "Author" in Figure 1), and the meta-information about each attribute (such as value type, default value, etc).
- *Web database clustering by topic:* Cluster all discov-

ered Web databases into different groups. The Web databases in each group belong to the same topic.

- *Interface integration:* Given the Web databases in the same topic, merge the same semantic attributes in different query interfaces into a global attribute, and finally form an integrated interface.

Query processing module: Process a user's query filled in integrated interface, and submit the query to each Web databases. There are three components in this module. The functions of them are described as following:

- *Web database selection:* Select appropriate Web databases for a user's query in order to get the satisfying results at minimal cost.
- *Query translation:* Try to translate the query on integrated interface equivalently into a set of local queries on the query interfaces of Web databases.
- *Query submission:* Analyze the submission approaches of local query interfaces, and submit each local query automatically.

Result processing module: Extract the query results achieved from Web databases, and merge the results together under a global schema. There are three components in this module. The functions of them are described as following:

- *Result extraction:* Identify and extract the pure results from the response pages returned by Web databases.
- *Result Annotation:* Append the proper semantics for the extracted results.
- *Result merging:* Merge the results extracted from different Web databases together under a global schema.

These components work together and make up of a comprehensive solution for Web database integration. It's not difficult to found that there are dependency relationships between them. Figure 2 has disclosed such dependency relationship. For example, query processing module depends on integrated interface generation module (high level), interface integration depends on Web database clustering (low level). So the quality of the implementation of a component will affect the next component greatly.

In fact, each component can be considered as a research issue itself. In order to build a practical Web database integration system, these issues must be solved well in theory first. In Section 3, the research status in this area will be discussed.

3. RESEARCH STATUS IN THIS AREA

Until now, large numbers of efforts are devoted to this area. Due to the space limit, the related works can not be discussed comprehensively and in detail. We only discuss them summarily according to the issues they address, and we also give the representative works.

Unfortunately, the development of research in this area is uneven very much though the great efforts have been done. Several issues have been already addressed well and are mature enough we can resort to (developed issues), some issues is developing and need be researched deeply (developing issues), and some issues have not been touched yet (undeveloped issues). We summarize the research status according to the development of these issues.

3.1 Developed Issues

Interface integration It has received enough attention, and several effective approaches[3][4][5][6] are proposed solve this problem. These approaches match attributes of query interfaces by exploiting the semantic similarity between labels as well as that between data instances.

Query interface schema extraction In order to understand query capabilities a query interface supports, [7] transforms query interfaces into a visual language, and develops a 2P grammar and a best-effort parser to realize a parsing mechanism.

3.2 Developing Issues

Besides introducing the current approach for developing issues, the shortcomings of them are pointed out at the same time.

Web database discovery [9] proposed a strategy does that by focusing the crawl on a given topic and choosing links to follow within a topic that are more likely to lead to pages that contain query interfaces. It can not assure the quantity of discovered Web databases. [10] use automatic feature generation to describe candidates and C4.5 decision trees to detect query interfaces. It can not differentiate the query interfaces of search engines from that of Web databases.

Web database clustering [11] performs the clustering based on the features available on the interface page. [12] proposed an objective function, model-differentiation, to compute the probability which topic a query interface belongs to. Their accuracy depends on the schema information of query interfaces, so they are not good at dealing with the query interfaces with simple schema.

Result extraction There are lots of approaches proposed to address this issue. Most of them[13][14][15] first transform the response page into a HTML tag tree, then identify and extract data records or data items by analyzing tree structure and tag information. They can only deal with the Web pages designed by HTML language, so it is a latent shortcoming with the development of Web.

Result annotation This problem is often solved during the process of Result extraction. [17] find the proper the annotation of an extracted data item in the response page by some heuristic rules. They are very effective if a data item really has its annotation in the response page. But they can not ensure all data items get their annotations.

Entity identification Entity identification is one of the key components of data merging. Several approaches have been proposed to solve this problem. For example, [16] applies a set of domain-independent string transformations to compare the entities' shared attributes in order to identify matching entities. All current approaches assume that they have achieved the well-build schema match between Web databases, but schema match in Web context have not been solved yet.

3.3 Undeveloped Issues

The undeveloped issues include *Web database selection*, *Query translation*, and *Data merging*. These issues have been well studied in some contexts(such as data warehouse), but there have not been approaches proposed to address these issues in the context of Web database integration, and they are compulsory in Web database integration.

Among these developing and undeveloped issues, *Entity identification*, *Result extraction* and *Web database selection* are in my PhD track at present and in the future, which are discussed in Section 4.

4. SEVERAL RESEARCH WORKS

In this section, several research works are proposed for discussion, which are being done at present and will be done in future.

4.1 Entity Identification among Web Databases

Entity identification is a key operation in integrating data from multiple sources. This issue has been well studied for years. As discussed in Subsection 3.2, though several solutions have already been proposed for Web databases, all of they are based on such assumption that the schema match between Web databases has been built well. As well known, due to the poor structure of Web pages, schema match in Web context is a very hard work, and there is still not automatic solution for it.

So we are trying to find a way to implement entity identification between Web databases without the help of schema match. Our basic consideration is described as following. We do not try to analyze the structure (or schema) of data records in response pages. Instead, given two Web databases A and B, each data record from A or B is considered as a text document. We judge whether data record a (from A) and data record b (from B) by comparing the text similarity of them. Obviously, it is very naive to compute the text similarity of two data records directly, and the accuracy is also not satisfying in our test. The reason is that, the importance of every part in a data record is different, and there is much noise information in a data record (for example, the words "author" and "price" often appear in the book data records). In order to make the similarity of a and b more reasonable (ideally, if a and b refer to a same entity, and a and c do not, then the similarity of a and b must be bigger than that of a and c), our approach is implemented as following:

1. filter the noise information from a and b as possible;
2. segment a into several blocks, and each block of a is formulated into a query for b;
3. compute the similarity of each block and b;
4. assign an appropriate weight for the similarity of each block and b, and sum up them;
5. judge whether a and b refer to a same entity according to the whole similarity.

At present, we are engaging to find an effective algorithm to train the weights and threshold of the whole similarity by a small set of sample data records pairs. A data record pair is two data records from different Web databases, and they refer to a same entity. The algorithm is now being detailed. The primary experiment result is very satisfying under the book topic. Further, the experiments under other topics (car, estate, etc.) will be done.

4.2 Vision Based Result Extraction

Most current approaches extract the results from response pages based on HTML language. But they have several inextirpable limitations. First, besides HTML, some other languages, such as XML and XHTML, have been introduced design Web pages. Second, HTML is still evolving. New versions of HTML will be proposed in the future, and new tags may appear and applied continuously. Third, as more and more web pages use more complex JavaScript and CSS to influence the structure of web pages, the applicability of the existing solutions will become lower. Fourth, if HTML is replaced by a new language in the future, then previous solutions will have to be revised greatly or even abandoned, and other approaches must be proposed to accommodate the new language.

Based on such motivations, it is important to find an approach which is vision based and language independent. In current phrase, we only aim at the response pages with multiple data records. Our basic idea is that, though the data records in a response page are different on the contents, they are similar on the appearance. The following is the implementation we are engaging in:

1. achieve the vision information (such as the font of a text, the size of an image, and their location in the Web page) by accessing the program interface of Web browser;
2. build a vision based block tree by VIPs[18] algorithm. A data record is composed by one or more blocks in the vision based block tree. So result extraction here is to find these blocks and judge which blocks compose a data record.
3. locate the data region (the region contains all data records in a response page) in the vision based block tree.
4. find the boundaries of all data records by computing the vision similarity of blocks in the vision based block tree.

The primary experiment has indicated that this approach is not only HTML language independent, but also very suit for extracting information-rich data records.

4.3 Web Database Selection

There are myriads of Web databases in the Web. So maybe a lot of Web databases are integrated under a topic. If a user submits a query on the integrated interface and the query is dispatched to all the Web databases integrated, it will be time-consuming and overhead to process all the returned results, especially data cleaning and deduplication. In most

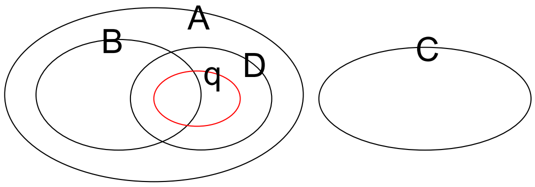


Figure 3: An example for Web Database Selection

cases, we only need select several ones among them to get the satisfying results. So *Web Database Selection* aims to select appropriate Web databases for a given user's query on integrated interface, which can help users get their desired results at the lowest cost.

In order to judge whether a Web database should be selected to answer a given query, there are two aspects must be considered. One is the pertinency of the Web database and the given query; the other is the query capability of the query interface of the Web database. The following gives some our considerations about the two aspects.

The prerequisite of selecting a Web database is it is pertinent to the given query. Extremely, it is meaningless to query a Web database if it does not has any useful information for the query. Figure 3 gives an example to illustrate this. Suppose A, B, C, and D are four Web databases, and q is a query to them. Where the size of A, B, C and D is the quantity of data records in them, the size of q is the quantity of data records satisfies q. Instinctively, C does not satisfy q at all, B satisfies q partly, A and D can satisfy q completely, but at last D is the best selection compared with A. So we need achieve the features of Web databases in advance. The features of a Web database include the size, the update ratio, the distribution on each attribute, etc. Because we can only access a Web database through its query interface, it is impossible to understand a Web database directly. The challenge is how to obtain the features by the query interface only. In the future, we want to design a sample records retriever to address this problem. Sample records retriever is a tool that can obtain a small set of data records which are distributed evenly in the Web database. We can profile the Web database by analyzing the obtained data records. Sample records retriever should have two components: query interface analyzer and query generator. Query interface analyzer is to obtain the necessary information of each attribute; query generator produces a set of smart queries according to the information obtained by query interface analyzer.

The query interfaces are often different about the query capability among Web databases, and this will influence the accuracy of a query. For example, in the book topic, a query on the integrated interface is "title=java and price<20$". If the query interface of a Web database contains both the two attributes , it can answer the query accurately. But if it only contains the attribute "title" or "price", then the results returned from the Web database will contain quite many data records which do not satisfy the query. So the challenge tasks are how to how to make the returned results be satisfying(for example, the minimal superset or maximal subset of the query).

5. CONCLUSIONS

With the rapid increasing of Web databases, it is impending to integrate these Web databases and provide people a unified access to them and achieve information automatically. In this paper, a comprehensive solution for Web database integration is proposed. There are a number of components in the solution, and each of them is also a research issue in this area. After summarizing the research statuses of the issues in this area, we introduce the issues which are being focused on now and will be addressed in the future. In conclusion, the focuses of my PhD thesis are building a Web database integration system and addressing several issues in this area.

6. REFERENCES

[1] K. C. Chang, B. He, C. Li, M. Patel, Z. Zhang. Structured Databases on the Web: Observations and Implications. SIGMOD Record 33(3): 61-70 (2004).

[2] http://www.completeplanet.com/.

[3] B. He, K. C. Chang. Statistical Schema Matching across Web Query Interfaces. SIGMOD Conference 2003: 217-228.

[4] H. He, W. Meng, C. T. Yu, Z. Wu. WISE-Integrator: An Automatic Integrator of Web Search Interfaces for E-Commerce. VLDB Conference 2003: 117-128.

[5] W. Wu, A. Doan, C. T. Yu. WebIQ: Learning from the Web to Match Deep-Web Query Interfaces. ICDE Conference 2006.

[6] E. Dragut, W. Wu, A. P. Sistla, C. T. Yu, W. Meng. Merging Source Query Interfaces on Web Databases. ICDE Conference 2006.

[7] Z. Zhang, B. He, K. C. Chang. Understanding Web Query Interfaces: Best-Effort Parsing with Hidden Syntax. SIGMOD Conference 2004: 107-118.

[8] H. He, W. Meng, C. T. Yu, Z. Wu. Automatic extraction of web search interfaces for interface schema integration. WWW Conference 2004: 414-415.

[9] L. Barbosa, J. Freire. Searching for Hidden-Web Databases. WebDB 2005: 1-6.

[10] J. Cope, N. Craswell, D. Hawking. Automated Discovery of Search Interfaces on the Web. ADC Conference 2003: 181-189.

[11] Q. Peng, W. Meng, H. He, C. T. Yu. WISE-cluster: clustering e-commerce search engines automatically. WIDM 2004: 104-111.

[12] B. He, T. Tao, K. C. Chang. Clustering Structured Web Sources: A Schema-Based, Model-Differentiation Approach. EDBT 2004: 536-546.

[13] B. Liu, R. L. Grossman, Y. Zhai. Mining data records in Web pages. KDD Conference 2003: 601-606.

[14] Y. Zhai, B. Liu. Web data extraction based on partial tree alignment. WWW Conference 2005: 76-85.

[15] H. Zhao, W. Meng, Z. Wu, V. Raghavan, C. T. Yu. Fully automatic wrapper generation for search engines. WWW Conference 2005: 66-75.

[16] S. Tejada, C. A. Knoblock, S. Minton. Learning domain-independent string transformation weights for high accuracy object identification. KDD Conference 2002: 350-359.

[17] J. Wang, F. H. Lochovsky. Data extraction and label assignment for web databases. WWW Conference 2003: 187-196.

[18] D. Cai, S. Yu, J. Wen, W. Ma. Extracting Content Structure for Web Pages Based on Visual Representation. APWeb Conference 2003: 406-417.